PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ^{6} :
H04L 12/66

A1

(11) International Publication Number: **WO 98/06204**

(43) International Publication Date: 12 February 1998 (12.02.98)

(21) International Application Number: PCT/US97/13943

(22) International Filing Date: 7 August 1997 (07.08.97)

(30) Priority Data:
08/694,076 8 August 1996 (08.08.96) US

(71) Applicant: AT & T CORP. [US/US]; 32 Avenue of the Americas, New York, NY 10013-2412 (US).

(72) Inventor: KILLIAN, Thomas, Joseph; 321 Saint Georges Place, Westfield, NJ 07090 (US).

(74) Agent: WEINICK, Jeffrey, M.; c/o Henry Brendzel, AT & T Corp., P.O. Box 4110, Middletown, NJ 07748 (US).

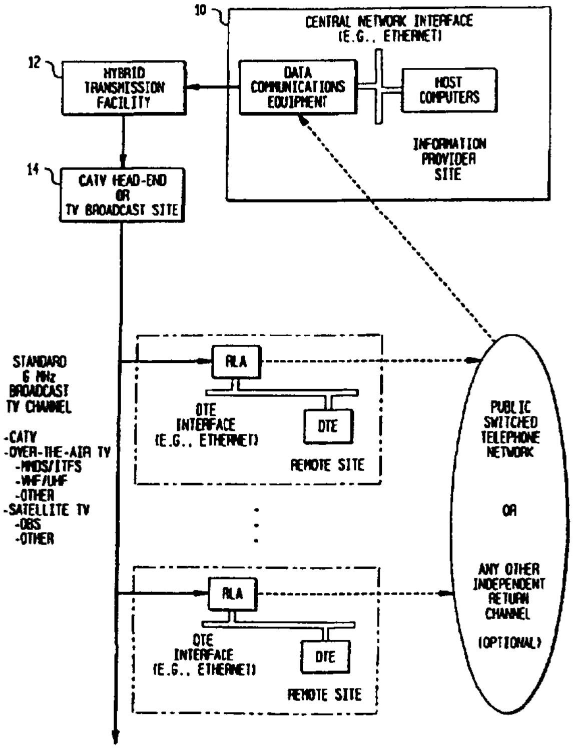
(81) Designated States: AU, CA, JP, MX, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: TRANSFERRING MESSAGES IN NETWORKS MADE UP OF SUBNETWORKS WITH DIFFERENT NAMESPACES

(57) Abstract

Techniques employed in packet networks for transferring a packet across subnetworks with different namespaces. When a packet enters a given subnetwork and has a destination in a subnetwork with a different namespace, the given subnetwork encapsulates the packet by adding a header which specifies a decapsulator in the namespace. When the packet arrives at the decapsulator, the decapsulator strips the header and provides the packet to a subnetwork with a different namespace. A particular use of the technique is in a network used for broad-band interactive service. The network has two subnetworks. The first subnetwork is a TV channel which functions as a high-bandwidth forward channel and the second subnetwork is a packet network accessible via a public modem pool which functions as a lower-bandwidth return channel. The encapsulator establishes a connection with the public modem pool and receives an address in the second subnetwork which is temporarily associated with the connection. When the encapsulator receives a packet which is produced in response to a packet received from the TV channel and has a destination address in the subnetwork of the TV channel, the encapsulator places a header on it which contains the temporary address and the address of the decapsulator. When the packet arrives at the decapsulator, the decapsulator removes the header and provides the packet to the second subnetwork.


10
CENTRAL NETWORK INTERFACE (E.G., ETHERNET)
12
HYBRID TRANSMISSION FACILITY
DATA COMMUNICATIONS EQUIPMENT
HOST COMPUTERS
INFORMATION PROVIDER SITE
14
CATV HEAD-END OR TV BROADCAST SITE
STANDARD 6 MHz BROADCAST TV CHANNEL
-CATV
-OVER-THE-AIR TV
-MMDS/ITFS
-VHF/UHF
-OTHER
-SATELLITE TV
-DBS
-OTHER
RLA
DTE INTERFACE (E.G., ETHERNET)
DTE
REMOTE SITE
RLA
DTE INTERFACE (E.G., ETHERNET)
DTE
REMOTE SITE
PUBLIC SWITCHED TELEPHONE NETWORK
OR
ANY OTHER INDEPENDENT RETURN CHANNEL
(OPTIONAL)

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece		Republic of Macedonia	**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's	**NZ**	New Zealand		
CM	Cameroon		Republic of Korea	**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

Transferring Messages in Networks made up of Subnetworks with Different Namespaces

Background of the Invention

Field of the Invention

The invention concerns transferring messages via networks generally and more particularly concerns transferring messages via networks in which there are subnetworks with different namespaces. An important area of application of the invention is high-bandwidth interactive services which use independent forward and return channels having different bandwidths.

Description of the Prior Art

An important problem in telecommunications is providing high-bandwidth interactive services to sites which do not have high-bandwidth two-way connections. An example of such a site is the average residential dwelling. A residential dwelling typically has two connections by which it may exchange data with the rest of the world. One of these is a connection to a cable TV (CATV) network. The other is a standard twisted pair telephone line. The CATV connection has high bandwidth, but may be used only to receive data, not send it. In the terminology used in this patent application, the CATV connection provides only a *forward* channel. The telephone line may be used to both send and receive data, and thus provides both a *return* channel and a forward channel, but both channels have low bandwidth.

Those who have studied the problem of providing high-bandwidth interactive services have noted that in many cases, the bandwidth required for the interaction is asymmetrical. One example is surfing the World Wide Web. The surfer sends the address of a Web page; the page is downloaded to the surfer's computer; then the surfer responds with the address of another web page, and so forth. Another example is telecommuting. There are many jobs in which the worker examines a document and does something with it such as modifying it, making a comment, or sending it to someone else and then gets the next document. In most cases, the amount of information output by the worker when he/she works on the document is much smaller than the amount of information contained in the document itself. Thus, what both the Web surfer and the telecommuter need is a high bandwidth forward channel upon which he/she can receive the web pages or documents to be examined and a low-bandwidth return channel for the information that he/she is outputting.

A system for providing interactive services that have a high-bandwidth forward channel and a low-bandwidth return channel to residential dwellings is described in Moura, et al, U.S. Patent 5,347,304, *Remote Link Adapter for use in TV Broadcast Data Transmission System*, issued Sept. 13, 1994. That entire patent is hereby incorporated by reference into the present patent application. FIGs. 1 and 2 of the present patent application are copies of the corresponding FIGS. of the Moura patent. As shown in FIG. 1, the system of Moura uses a CATV or broadcast TV channel as the high-bandwidth forward channel and has an independent low-bandwidth return channel. The return channel may be a circuit in the public switched telephone network or any other channel which is independent of the forward channel. Information provider site 10 and each of the remote sites has an address in the network formed by the forward channel, the return channel, the information provider site, and the remote sites.

The data in the forward channel comes from host computers in information provider site 10; the data communications equipment makes *packets* containing the data. Each packet includes a *message* which contains the data and a *header*. The header contains at least the address of the source of the message and the address of the destination of the message. In this case, the source address is the address of information provider site 10 and the destination address is the address of one of the remote sites. The packet goes from data communications equipment to hybrid transmission facility 10, which employs a radio frequency modem to convert the digital packet into a form proper for its transmission over the TV channel and provides it to head end or broadcast site 14. At each remote site, a remote link adapter (RLA) which includes another radio frequency modem watches the TV channel for packets addressed to its remote site. When it detects such a packet, it converts the packet into a form proper for its transmission over the network connecting data terminal equipment (DTE) such as a personal computer and the RLA and provides the packet to the DTE. As can be seen from the foregoing, the forward channel is in fact a packet network, and will be termed in the following a *forward packet network*.

When the user of the DTE responds to the data that has been sent him/her via the forward packet network, the response goes to the RLA, which sends it via the telephone line in the user's dwelling and the public switched telephone network to data communications equipment at information provider 10, which in turn provides the response to the host computers at information provider site 10.

The Moura patent provides substantially no disclosure about the return channel. One particularly useful form of return channel is a packet network. Such a return channel is termed in the following a *return packet network*. When the return channel is a return packet network, the packets that contain the response must include a header with the address of the remote site as a source address and the address of the data provider as a destination address. This is all straightforward enough, but it necessarily assumes that the forward packet network and the return packet network have the same set of addresses. If they do not, the addresses necessary to communicate with sources and destinations in the forward packet network must be added to the return packet network.

Adding addresses may be difficult for several reasons:

- if there is a large number of provider sites and remote sites, a substantial increase in the size of the return packet network's address data bases will be required;

- considerable time must be spent updating the return packet network's address data bases with the new names; and

- the return packet network's address data bases must track changes in remote sites and data provider sites in the forward packet network.

These problems are particularly difficult if the return packet network is not under control of the entity that controls the forward packet network. Precisely that is the case when a public packet network such as those made available by internet access providers is used as the return packet network. Using such a public packet network as the return packet network in the system of Mourra is however particularly advantageous, first, because the necessary infrastructure already exists, and second because the competition of the public packet networks for traffic reduces the cost of the return channel to the user.

The need to add addresses to the return packet network is a specific example of a general problem with packet networks: each packet network has a *namespace*, that is, a set of addresses that it recognizes as sources of or destinations for packets. If a packet network receives a packet whose source or destination address is not part of the namespace of the network, the message is not forwarded. Because that is the case, a packet network cannot include subnetworks (such as the forward packet network and return packet network in the implementation of Moura described above) that have different namespaces. When a packet which originated in a first subnetwork with a first namespace enters a second subnetwork with a second namespace, the second subnetwork will reject the packet. Consequently, a packet with source and destination names from a subnetwork having a first namespace cannot be sent via a subnetwork having a second namespace. As described above, the only way of presently solving this problem is to add the source and destination addresses to the second namespace.

It is an object of the invention disclosed herein to provide techniques for sending a packet via a path that crosses packet subnetworks with different namespaces.

Summary of the Invention

The object of the invention may be attained by the following method:

- when the packet enters a given one of the subnetworks and a current header of the packet does not specify a destination in the given one of the subnetworks, adding a new header to the packet and treating the new header as the current header, the new header specifying an exit destination where packets transferred via the path enter another one of the subnetworks;

- sending the packet to the destination specified in the current header; and
- if the packet arrives at the exit destination, removing the current header and providing the packet to the other subnetwork.

The portion of the method which adds the new header is termed *encapsulation*, and may be performed in apparatus called an *encapsulator*. The portion of the method which removes the current header is termed *decapsulation* and may be performed in apparatus called a *decapsulator*.

One application of the technique is in packet networks having a first subnetwork as a forward channel and a second subnetwork as a return channel. The encapsulator receives packets with addresses in the first subnetwork, appends the new header with the address of the decapsulator in the second subnetwork, and provides the packet to the second subnetwork. The second subnetwork sends the packet to the decapsulator, which removes the new header and provides the packet to the first subnetwork.

In a particular application of the technique, the forward channel is a TV channel and the return channel is a packet network to which access may be had by means of a public modem pool. When a user is accessing the packet network by means of a modem in the public modem pool, a temporary address in the packet network is associated with the modem. In this application, the encapsulator has a connection to a modem in the public modem pool, and has received the temporary address associated with the modem. The header added by the encapsulator includes the temporary address and the address of the decapsulator. The encapsulator may receive the address of the decapsulator via either the first packet network or the second packet network.

Other objects and advantages of the apparatus and methods disclosed herein will be apparent to those of ordinary skill in the art upon perusal of the following Drawing and Detailed Description, wherein:

Brief Description of the Drawing

FIG. 1 shows the prior-art system of Moura. It is a copy of FIG. 1 of the Moura patent;
FIG. 2 is a detail of Moura's Remote Link Adapter. It is a copy of FIG. 2

of the Moura patent;

FIG. 3 is a conceptual view of the invention;

FIG. 4 shows how the invention would be implemented in the system of Moura;

FIG. 5 shows outer header 317 in a preferred embodiment;

FIG. 6 is a diagram of subnetworks with nested namespaces;

FIG. 7 is a diagram of subnetworks with namespaces that are not nested;

FIG. 8 is a diagram of entry points encorporating the principles of the invention;

FIG. 9 is a detail of the Remote Link Adapter when it is configured to be an encapsulator; and

FIG. 10 is source code in the C language for an implementation of a decapsulator.

Reference numbers in FIGs. 1 and 2 have two digits; the reference numbers in FIGs. 3-9 have three digits: the two least-significant digits are the number of an item in a figure; the remaining digits are the number of the figure in which the item first appears. Thus, an item with the reference number 301 first appears in FIG. 3.

Detailed Description of a Preferred Embodiment

The following *Detailed Description* begins with a description of a technique called *internet protocol tunneling*, which is related to the technique used in the present invention to permit a message from a source in a first namespace to be transferred via a second namespace to a destination in the first namespace. Then a conceptual overview of the invention will be presented, followed by a description of an embodiment of the invention which serves as the return channel in the system of Moura.

Internet Protocol Tunneling

Internet protocol tunneling is a technique which has long been used in the Internet to bridge portions of the Internet which have disjoint capabilities or policies. For instance, some portions of the Internet are surrounded by firewalls, that is, packets passing from outside the portion surrounded by the firewall to inside the portion and vice-versa are checked to determine whether they are authorized, and are allowed inside the portion only if they are. Tunneling may be used to permit packets that are not authorized to enter the portion surrounded by the firewall to to pass *through* the portion without interacting with anything inside the portion.

The actual tunneling protocol used in the to implement the invention is described in W. Simpson, *IP in IP Tunnelling*, Network Working Group Request for Comments: 1853, October, 1995. As set forth in the above reference, tunneling is done as follows: in the network node that marks the beginning of the portion of the network that is to be tunneled through, an *encapsulator* implemented in the node adds an *outer header* to the packet whose source address is that of the encapsulator and whose destination address is that of a *decapsulator* in a node that marks the end of the portion of the network that is to be tunnelled through. The portion of the network being tunnelled through reads only the *outer header* and simply sends the packet to the decapsulator. The decapsulator removes the outer header and provides the original packet to the next portion of the network. The outer header thus *encapsulates* the original packet as it passes through the tunnel. Of course, it may turn out that while a packet is in one tunnel, it must pass through another tunnel. In that situation, the encapsulator for the new tunnel simply places a new outer header at the head of the packet, and the decapsulator for the new tunnel removes the new outer header when the end of the new tunnel is reached. There is of course no limit to the number of times this technique may be repeated.

Using Tunneling to Route a Packet through a Subnetwork with a Different Namespace: FIG. 3

In the present invention, the technique of tunneling is put to a new use: namely to route a packet through subnetworks that have different namespaces than the one to which the source and destination addresses in the packet's original header belong. The new use, termed herein *namespace tunneling*, is shown in FIG. 3. There, a network 301 is shown which has a subnetwork 303 with namespace A and a subnetwork 325 with namespace B. A packet 305 with a source whose address belongs to namespace A must travel via subnet 325 to a destination whose address belongs to namespace A. Packet

305 has message 311 and inner header 306 which contains subnetwork A source address (NAS) 307 and subnetwork A destination address (NAD) 309. Since packet 305 must travel via subnetwork B, a router in subnetwork A sends packet 305 to encapsulator 313, which may receive packets from subnetwork A and provide them to subnetwork B. Encapsulator 313 has in its memory its own address in subnetwork B and the address of decapsulator 315 in subnetwork B, as well as any other information needed to properly construct an outer header. Encapsulator 313 receives packet 305 and adds to it outer header 317, which includes at least the address in subnetwork B of encapsulator 313 as the source address and the address in subnetwork B of decapsulator 315 as the destination address. Encapsulated packet 323 is now sent by subnetwork B to decapsulator 315, which has access to subnetwork A. Decapsulator 315 simply removes outer header 317 and delivers packet 305 to subnetwork A, which in turn delivers the packet to the destination specified in NAD field 309.

Namespace Tunneling with more than one Subnetwork having a different Address Space: FIGS. 6-8

Of course, namespace tunneling may be used where the packet is transferred through more than one namespace. In describing tunneling in this case, it is useful to define the concept of a packet's *path* through the namespaces. For purposes of the present discussion, the path of a packet is simply the order in which the packet encounters the namespaces. There are two kinds of orders: nested order and sequential order.

FIG. 6 shows a path with nested order. There are three namespaces C, belonging to subnetwork 603, D, belonging to subnetwork 605, and E, belonging to subnetwork 607. Packet 305 starts in namespace C, enterns namespace D, then enters namespace E, then enters D again, and finally C again. The path is C,D,E,D,C. Before packet 305 enters namespace D, it has the form shown at 608; Subnetwork 603 routes it to encapsulator 609, which makes packet 610 by adding an outer header 317'. Outer header 317' includes the address of decapsulator 619. Subnetwork 605 routes the packet to subnetwork 607, which makes packet 617 by adding another outer header 317", which contains as its destination address the address of decapsulator 615. When decapsulator 615 receives packet 613, it removes outer header

317" to produce packet 617, which, as specified in outer header 317', is routed by subnetwork 605 to decapsulator 619, which removes outer header 317' to produce packet 621 in subnetwork 603, which of course contains inner header 306 of the original packet 608. As will be immediately apparent, the nesting of address spaces may be to any practical depth. As will also be immediately apparent, when a namespace is nested in another namespace, any path out of the subnetwork to which the nested namespace belongs must enter the other namespace, and consequently, the nested namespace requires only one decapsulator 619.

In the other, shown in FIG. 7, the path taken by packet 709 again enters three namespaces, namespace F belonging to subnetwork 303, namespace G belonging to subnetwork 705, and namespace H belonging to subnetwork 707. The path this time is F,G,H,F. In this path, namespaces G and H are nested in F, but neither G nor H is nested in the other. As packet 709 begins traveling via the path, it is in namespace F and contains only the message and the inner header, with source and destination addresses in namespace F. Since the path goes by G and H, the packet 709 goes to encapsulator 711, which adds header 317' to make packet 713. Header 317' contains the address of decapsulator 715, which removes outer header 317' to produce packet 717. That in turn is routed to encapsulator 719, which adds outer header 317" to produce packet 707, with outer header 317" specifying decapsulator 723 as a destination. Decapsulator 723 strips header 317" to produce packet 725, which is identical with packet 709.

When there are non-nested namespaces, a given subnetwork must contain a decapsulator for each subnetwork which immediately follows the given subnetwork in the path. In that case, a path for a given packet may be established prior to transmitting the packet by dynamically setting the encapsulators for the packet's path to provide the addresses for the proper decapsulators prior to sending the packet. A path may also be established or the packet may by include routing information from which the encapsulator can determine which decapsulator the packet should be directed to.

General Method of Namespace Tunneling

There is a general method of namespace tunneling which works for paths which have nested namespaces, sequential name spaces, or combinations of the two. In describing the method, it is useful to employ the term *current* *header*. The current header for a packet is the one used to route the packet in the namespace of the subnetwork that the packet is currently traveling in. Thus, when packet 721 is traveling in subnetwork 707, outer header 317'' is the current header; when packet 721 is travelling in subnetwork 703, the inner header is the current header.

The method is the following:

- When the packet is received in a given one of the subnetworks and a current header of the packet does not specify a destination in the given one of the subnetworks, add a new header to the packet and treat the new header as the current header. The new header specifies an exit destination where packets transferred via the path enter another one of the subnetworks.
- Send the packet to the destination specified in the current header.
- If the packet arrives at the exit destination, remove the current header and provide the packet to the other subnetwork.

In terms of the foregoing discussion, the encapsulator is located at the point at which the packet is received in the given subnetwork and the decapsulator is located at the exit destination.

Implementation of the Method

One way of setting up encapsulators and decapsulators to implement the foregoing method of namespace tunneling is shown in FIG. 8, which shows two entry points 801 between a subnetwork 303 with a namespace A and a subnetwork 325, with a namespace B. In this arrangement, there are two entry points, A to B entry point 815, which permits packets arriving from subnetwork 303 to enter subnetwork 325, and B to A entry point 817, which does the reverse. Only entry point 815 will be described in detail, since entry point 817 works in exactly the same way, but in the reverse direction. Of course, a subnetwork with a given namespace must have entry points for all of the subnetworks with other namespaces to which the subnetwork with the given namespace either directly provides packets or from which the subnetwork with the given namespace directly receives packets.

The only packets which subnetwork 303 will route to decapsulator 803 is those for which the current outer header 317 contains the address in namespace A of decapsulator 803. Decapsulator 803 will always remove current outer header 317 and pass the packet minus current outer header 317 to encapsulator 805. What encapsulator 805 does depends on whether the header which follows the header that was stripped specifies source and destination addresses in namespace B. If the header does, encapsulator 805 does not add a new outer header 317 to the packet, but simply provides the packet to subnetwork 325, which transfers it to the destination specified in the header. If the header specifies addresses in a namespace other than namespace B, say namespace X, encapsulator 805 adds new outer header 317 with the address in namespace B of the decapsulator for the B to X entry point.

The first case is shown with packet 807, which has an inner header 306 that specifies a source and destination in namespace 325, and has an outer header 317 that specifies decapsulator 803. Decapsulator 803 strips outer header 317; encapsulator 805 reads inner header 306 and determines that the addresses in inner header 306 are in namespace B, so it simply provides packet 809 to subnetwork 325. It should be noted here that encapsulator 805 would have done the same thing had inner header 306 been an outer header 317 with addresses in namespace B.

The second case is shown with packet 811, which has an inner header 306' that specifies names in a namespace X of another subnetwork. As before, decapsulator 803 strips outer header 317; however, when encapsulator 805 reads inner header 306', it determines that it does not contain addresses in namespace B, so it adds a new outer header 317' which contains the address in namespace B of decapsulator 803 for the subnetwork with namespace X. Again, encapsulator 805 would have done the same thing had inner header 306' been an outer header 317 with addresses in namespace X.

As will be apparent from the foregoing discussion, the path that a given packet takes through the namespaces is determined by the destination addresses that the encapsulators put in the outer headers. If all of the encapsulators are under control of the same entity, that entity can determine the destination addresses and can thus define paths; otherwise, consensual arrangements for defining paths must be made among the users of the network.

Using Namespace Tunneling in Broadband Interactive Systems: FIGs. 1,2, 4, and 5

As set forth in the *Description of the Prior Art*, the return channel broadband interactive systems such as that of Moura may be implemented using a packet network which is accessible by means of a public modem pool. Such a packet network is termed herein a *public packet network*. Examples of such packet networks are those belonging to on-line service providers. In such implementations, the public packet network and the packet network which implements the forward channel have different namespaces. As indicated above, the necessary names can be added to the namespace of the public packet network, but doing so substantially increases the cost and complexity of maintaining the namespace of the public packet network. Namespace tunneling may be used to solve this problem.

FIG. 4 shows an implementation 401 of the system of Moura in which a public packet network belonging to an on-line service provider 407 provides the return channel. Implementation 401 forms a network 402 which has two subnetworks: subnetwork 404, which is the packet network of the forward channel, and subnetwork 406, which is the public packet network of the return channel. Subnetwork 404 has namespace A 411 and subnetwork 406 has namespace B 413. As set forth above, absent namespace tunneling, source and destination names from namespace A 411 must be added to namespace B 413 if RLA 201 is to be able to send responses to packets received in RLA 201 from subnetwork 404 to be returned via subnetwork 406 to a destination in subnetwork 404.

When namespace tunneling is used in system 401 of Mourra, RLA 201 functions not only as a node in subnetwork 404, but also as an encapsulator 313 for subnetwork 406. A device which had an address in namespace B and access to subnetwork 404 functions as a decapsulator 321. Operation of implementation 401 is as follows: as before, information provider site 10 provides digital packets with source and destination addresses in subnetwork 404 to hybrid transmission facility 12, which puts the packets into a form such that they can be broadcast on a TV channel from CATV head end 14. RLA 201 has an address in subnetwork 404 and monitors the packets transmitted on the TV channel. When one is transmitted with is addressed to RLA 201, RLA 201 converts it to the proper form for data terminal equipment 403 and provides it to data terminal equipment 403. Information in the packet, together with information from other packets, is used to create a display on data terminal equipment 403 to which the user of data terminal equipment 403 may respond. If the user does, data terminal equipment 403 sends a packet containing the response to RLA 201. The packet has the address of RLA 201 in subnetwork 404 as its source address and the address of information provider 10 in subnetwork 404 with which the user is presently interacting. In the terms used in the general discussion of namespace tunneling, the packet is a packet 305 which has not been encapsulated.

RLA 201 has access via public switched telephone network 408 to a public modem pool 409 belonging to on-line service provider 407. When the user of data terminal equipment 403 is interacting with information provider site 10, there is a circuit 405 in telephone network 408 between a modem in modem pool 409 and a telephone modem in RLA 201 (see hybrid interface 22 in FIG. 2). According to the standard practice of on-line service providers, at the time circuit 405 is established, on-line service provider 407 assigns the circuit a temporary address in subnetwork 406 and returns the temporary address to RLA 201, which stores the temporary address in its memory. RLA 201 further has stored in its memory the address in subnetwork 406 of decapsulator 321. This address may be built into RLA 201, or RLA 201 may have received the address in a message which it receives from either subnetwork 404 or subnetwork 406. For example, if system 401 is being used for telecommuting, information provider site 10 will typically belong to the user's employer, and that employer may have selected a preferred on-line service provider for the return channel. In such a case, the message with the address of decapsulator 321 would be sent form information provider site 10 via subnetwork 404.

When RLA 201 receives packet 305 from DTE 403, it adds an outer header 317 to packet 305 to create an encapsulated packet 323. Outer header 317 contains the temporary address associated with circuit 405 as the source address in subnetwork 406 and the address of decapsulator 321 as the destination address. RLA 201 then sends encapsulated packet 323 via telephone circuit 405 to on-line service provider 407. On-line service provider 407 accepts packet 323 because outer header 317 specifies a source and destination in namespace B 413 and forwards packet 323 to decapsulator 321, as specified in the destination address in outer header 317. Decapsulator 321 strips outer header 317 from encapsulated packet 323, leaving packet 305, which it it provides to subnetwork 404. The source address in inner header 307 is that of RLA 201 in subnetwork 404, and the destination address is that of information provider site 10, and so subnetwork 404 forwards the packet to information provider site 10.

As can be seen from the foregoing, when namespace tunneling is used in network 402, the only change that need be made in subnetwork 406 is the establishment of a decapsulator 321 at an address in subnetwork 406. There is no need to add addresses from subnetwork 404 to subnetwork 406 or to maintain any consistency between the networks beyond the address of decapsulator 321. Implementation of decapsulator 321 also poses no difficulty. The only packets which decapsulator 321 receives are those with an outer header 317 that specify the decapsulator in its destination address field, and all decapsulator 321 does is strip the outer header 317 from the packet and provide it to subnetwork 404.

Details of the Implementation of Encapsulator 313 in RLA 201: FIGs. 2, 5, 9, and 10

FIG. 2 is from the Moura patent. The figure shows details of RLA 201 in the system of Moura. The three components of RLA 201 are user interface 20, which provides the connection to DTE 403, hybrid interface 22, which provides the connection via modems to the CATV channel and to the switched telephone network, and engine 24, which contains a microprocessor which controls the operation of RLA 201 and the ROM and RAM needed to store programs and data for the microprocessor. When an encapsulator 313 is implemented in RLA 201, RLA 201 is modified as shown in FIG. 9. That Figure shows parts of the contents of memory 901 of engine 24. Memory 901 has two components: RAM 915 and ROM 915. In the implementation, RAM 902 contains outer header information 903 which the microprocessor uses to make outer header 903. Included in outer header information is namespace B source address 905 for the outer header and namspace B destination address 907 for the outer header. As explained above, address 905 is the temporary address given circuit 405 by on-line service provider 407 and address 907 is the address of decapsulator 321. ROM 915 contains the code necessary to implement the encapsulator. Encapsulator code 911 uses outer header information 903 to make encapsulated packets 323 from packets 305; downloading code 913 contains the code which downloads outer header information including the address 907 of decapsulator 321 from either subnetwork

404 or subnetwork 406. Decapsulator 321 may be similarly implemented by writing code for decapsulator 321 and executing the code in a processor at the decapsulator's address in subnetwork 406.

FIG. 10 shows code 1001 for a presently-preferred embodiment of decapsulator 321. In the computer system on which code 1001 is executed, a daemon (a process that is continually active) watches the destination addresses of packets as they pass on an ethernet. The daemon has a list of destination addresses for which it reads packets and it also can associate with each address a program which specifies how a packet directed to that address is to be processed. When the daemon sees a packet with a destination address which is on its list, it reads the packet, processes it as specified in the program, and sends it to the process corresponding to the destination. In the case of packets directed to the destination address at which the decapsulator resides, the program associated with that address removes outer header 317 from the packet.

The process executing Code 1001 provides the stripped packets it receives from the daemon to subnetwork 404. Normally, when an entity sends a packet, it writes its address into the packet as the packet's source address; here, however, the source address cannot be written over, because it is the address of the source in subnetwork 404. At 1003, the code sets options that prevent the source address from being overwritten. The `while` loop at 1005 simply reads the packets provided by the daemon on `stdin` and sends them unaltered to subnetwork 404.

FIG. 5, finally, shows outer header 317 in a preferred environment. FIG. 5 also shows the significance of the fields. In the description of the fields, the "user's packet" is a packet 306; the value of Protocol field 517 identifies the data structure as and outer header 317; the header checksum is used for error detection in the header; and the dialup address is the temporary address assigned to circuit 405 by on-line service provider 407.

Conclusion

The foregoing *Detailed Description* has disclosed to those skilled in the arts to which the invention pertains how one may use namespace tunneling to transfer packets across packet networks that have subnetworks with different namespaces and has further disclosed how namespace tunneling may be used in the context of a system for providing interactive broadband service to residences to make it easier to use a public packet network as a return channel. The *Detailed Description* has further disclosed the best mode presently known to the inventor of implementing his invention. However, as will be immediately apparent to those skilled in the arts to which the invention pertains, the principles of the invention have many possible embodiments. For example, the preferred embodiment uses the IP tunneling protocol described by W. Simpson; however, any encapsulating protocol that permitted specification of source and destination addresses for the encapsulated packet in an outer header and provided that any other source and destination addresses in the encapsulated packet would be ignored would work also. There are further many ways of implementing encapsulators and decapsulators; all that is necessary is that the encapsulator be able to add an outer header with the address of a decapsulator and the decapsulator be able to strip the outer header and provide the packet to the next address space. Moreover, there are many different ways of providing the address of the next decapsulator to the encapsulator.

With regard to the use of namespace tunneling to make it easier to use a public packet network as a return channel in systems like those of Moura, it should be pointed out that tunneling can be used wherever the return channel and the forward channel have different namespaces. It is thus by no means limited to the system of Moura or even to interactive broad-band systems with asymmetrical return channels generally.

All of the above being the case, the foregoing Detailed Description is to be understood as being in every respect illustrative and exemplary, but not restrictive, and the scope of the invention disclosed herein is not to be determined from the Detailed Description, but rather from the claims as interpreted according to the full breadth permitted by the law.

What is claimed is:

Claims:

1. A method employed in a packet network to transfer a packet via a path that involves subnetworks having different namespaces, the method comprising the steps of:

when the packet enters a given one of the subnetworks and a current header of the packet does not specify a destination in the given one of the subnetworks, adding a new header to the packet and treating the new header as the current header, the new header specifying an exit destination where packets transferred via the path enter another one of the subnetworks;

sending the packet to the destination specified in the current header; and

if the packet arrives at the exit destination, removing the current header and providing the packet to the other subnetwork.

2. The method set forth in claim 1 wherein:

when the packet enters the given one of the subnetworks, the current header of the packet never specifies a destination in the given one of the subnetworks.

3. The method set forth in claim 2 wherein:

the path is such that the namespace of the given one of the subnetworks is nested in the namespace of the other subnetwork.

4. The method set forth in any of claims 1 through 3 further comprising the step of:

determining the exit destination prior to entry of the packet into the given one of the subnetworks.

5. The method set forth in any of claims 1 through 3 further comprising the steps of:

determining the other subnetwork by means of routing information in the packet; and

setting the exit destination as required for the other subnetwork that was so determined.

6. A method, practiced in a first subnetwork of a network wherein there are subnetworks with different namespaces, of transferring a packet through the first

subnetwork to a first destination in a second subnetwork with a different namespace, the packet having a first header specifying the first destination and having been encapsulated on entry into the first subnetwork with a second header specifying a second destination in the first subnetwork,

the method comprising the steps of:

receiving the encapsulated packet at the second destination;

removing the second header; and

providing the packet to the second subnetwork.

7. A method of commencing transfer of a packet through a first subnetwork of a network to a second subnetwork thereof which has a namespace which is different from the namespace of the first subnetwork,

the method comprising the steps of:

encapsulating the packet with a header specifying a destination in the first subnetwork at which the header will be removed and the packet provided to the second subnetwork; and

providing the encapsulated packet to the first subnetwork.

8. The method set forth in claim 7 further comprising the step of:

determining the destination in the header prior to receiving the packet.

9. The method set forth in claim 7 further comprising the step of:

determining the destination in the header on the basis of routing information in the packet.

10. A data storage device characterized in that:

the data storage device contains code which when executed performs the method set forth in any of claims 6, 7, 8, or 9.

11. An improved subnetwork of a packet network that has another subnetwork with a different namespace, the subnetwork having the improvement comprising:

a first destination in the subnetwork which has access to the other subnetwork and which responds to a packet with a second destination that is in the other subnetwork and which was encapsulated upon entry into the subnetwork with a header containing the address of the first destination by removing the header from the packet and providing the packet to the other subnetwork.

12. Apparatus employed in a first subnetwork of a network wherein there are subnetworks with different namespaces to transfer a packet through the first subnetwork to a first destination in a second subnetwork with a different namespace, the packet having a first header specifying the first destination and having been encapsulated on entry into the first subnetwork with a second header specifying a second destination in the first subnetwork,
the apparatus comprising:

means at the second destination for receiving the packet;

means for removing the second header; and

means for providing the packet to the second subnetwork.

13. Apparatus for commencing transfer of a packet through a first subnetwork of a network to a second subnetwork thereof which has a namespace which is different from the namespace of the first subnetwork,
the apparatus comprising:

means for receiving the packet;

means for encapsulating the packet with a header specifying a destination in the first subnetwork at which the header will be removed and the packet provided to the second subnetwork; and

means for providing the encapsulated packet to the first subnetwork.

14. A method of transferring a packet having first source and destination addresses in a first subnetwork of a packet network through a second subnetwork thereof that has a different namespace from that of the first subnetwork to the location specified by the first destination address,
the method comprising the steps of:

on entry of the packet into the second subnetwork, encapsulating the packet with a header which has a second source address in the second subnetwork and a second destination address indicating a point where the packet leaves the second network;

sending the packet to the second destination address; and

at the second destination address, removing the header and providing the packet to the first subnetwork.

15. The method set forth in claim 14 further comprising the step of:

determining the second destination address prior to arrival of the packet at the point where the packet enters the second network.

16. A packet network comprising:

a first subnetwork having a first name space;

a second subnetwork having a second namespace;

an encapsulator which has access to the second namespace for receiving a packet from the first subnetwork which has a first destination in the first subnetwork and encapsulating the packet with a header specifying a second destination in the second subnetwork; and

a decapsulator at the second destination which has access to the first subnetwork, the decapsulator operating to remove the header from the packet and provide the packet to the first subnetwork.

17. The packet network set forth in claim 16 further comprising:

means in the encapsulator for receiving a message specifying the second destination.

18. The packet network set forth in claim 17 wherein:

the means for receiving the message receives the message in the alternative from the first network and the second network.

19. An interactive system comprising:

a high-bandwidth packet network which serves as a forward channel, the high-bandwidth packet network having a first namespace;

a lower-bandwidth packet network which serves as a return channel, the lower-bandwidth packet network having a second namespace;

interactive means having an address in the high-bandwidth packet network for receiving a first packet of data from the high-bandwidth packet network and providing a second packet in response thereto which has a first destination in the high-bandwidth network;

an encapsulator which is coupled to the interactive means and has access to the lower-bandwidth packet network for encapsulating the second packet with a header specifying a second destination in the lower-bandwidth packet network and providing the packet to the lower-bandwidth packet network; and at the second destination, a decapsulator which has access to the high-bandwidth packet network for removing the header from the second packet and providing the second packet to the high-bandwidth packet network.

20. The interactive system of claim 19 wherein:

the lower-bandwidth packet network includes means for temporarily assigning a temporary address in the lower-bandwidth packet network to a user thereof; and

the encapsulator means further specifies the temporary address as a source in the header.

21. The interactive system of claim 20 further comprising:

a connection between the encapsulator means and the lower-bandwidth packet network including a first modem, a circuit in a public switched telephone system, and a modem in a public modem pool which provides access to the lower-bandwidth network; and

wherein the means for temporarily assigning assigns one of the temporary addresses to the connection and provides the assigned temporary address to the encapsulator via the connection.

22. The interactive system of claim 19 wherein:

either the high-bandwidth packet network or the low-bandwidth packet network provides the second destination to the encapsulator.

23. A packet network of the type which transfers packets having headers which specify source addresses and destination addresses in the packet network, the packet network being characterized by:

a decapsulator at an address in the network, the decapsulator operating to remove the header from a packet having the address of the decapsulator and provide the packet to another packet network having a different namespace, and

a source of the packet external to the packet network;

and wherein

the packet network provides a temporary address in the packet network to the source the of packet and receives therefrom the packet, the packet having a header which specifies the temporary address as a source address and the address of the decapsulator as a destination address and the packet network routes the packets from the source to the decapsulator.

24. The packet network of claim 23 further characterized by:

a connection to the source of the packet which includes a modem in a public modem pool and a circuit in a public switched telephone system, the temporary address being assigned to the connection.

25. The packet network set forth in either of claims 23 or 24 further characterized in that:

the packet network further provides the address of the decapsulator to the source of the packet.

26. An encapsulator for encapsulating packets received from a first packet network with a first namespace and providing the packets to a second packet network with a second namespace,

The encapsulator comprising:

means for storing a source address in the second namespace and a destination address in the second namespace, the destination address being that of a decapsulator, the decapsulator operating to remove a header from a packet having the address of the decapsulator and provide the packet to the first packet network;

means for adding a header containing the source address and the destination address to the packet; and

means for providing the packet with the header to the second packet network.

27. The encapsulator set forth in claim 26 wherein:

the mean for providing the packet with the header to the second packet network includes

a modem capable of being coupled to a circuit in a public switched telephone network which is in turn coupled to a public modem pool belonging to the second packet network.

28. The encapsulator set forth in claim 27 wherein:

the second packet network temporarily associates an address in the second packet network with the circuit; and the source address is the temporarily-associated address.

29. the encapsulator set forth in any of claims 26 through 28 wherein: the encapsulator receives the destination address from either the first packet network or in the alternative, from the second packet network.

30. A data storage device characterized in that: The data storage device contains code which, when executed on a processor, implements the apparatus set forth in any of claims 26, 27, or 28.

31. Improved apparatus for receiving first packets from a high-bandwidth forward channel belonging to a first packet network and providing second packets to a return channel belonging to a second packet network accessible via a circuit in a public switched telephone network, the first and second packet networks having different namespaces,

the apparatus including

a first interface to the forward channel,

a second interface to the return channel,

a third interface to apparatus which receives the first packets and provides the second packets, and

control apparatus including a microprocessor and memory containing programs and data used by the microprocessor, the microprocessor operating under control of the programs and data to control the first, second, and third interfaces,

and the improvement comprising:

a first source address in the second packet network and a first destination address in the second packet network stored in the memory, the destination address being that of a decapsulator in the second packet network, the decapsulator operating to remove a header from a packet having the address of the decapsulator and provide the packet to the first packet network; and

a first program stored in the memory means which, when executed by the microprocessor, causes an additional header containing the first source address and the first destination address to be added to a packet received from the third interface that is to be sent via the the second interface and the second packet network to a second destination address in the first packet network.

32. The apparatus set forth in claim 31 further comprising:

a second program stored in the memory which is executed by the microprocessor in response to a packet received via either the first or second packet network that contains a message specifying an address of a decapsulator and which, when executed, stores the address of the decapsulator in the memory as the first destination address.

33. The apparatus set forth in claim 31 or claim 32 further comprising:

a third program which is executed by the microprocessor in response to establishment of the circuit in the switched public telephone network and which, when executed, receives an address associated with the circuit from the second packet network and stores that address in memory as the first source address.

34. A method of encapsulating a packet which travels via a second packet network with a second namespace to a destination in a first packet network with a first namespace,

the method comprising the steps of:

receiving the packet in an encapsulator which has access to the second packet network; and

adding a header to the packet which contains a source address and a destination address in the second namespace to the packet prior to providing the packet to the second namespace, the destination address being that of a decapsulator and the decapsulator operating to remove a header from a packet having the address of the decapsulator and provide the packet to the first packet network.

35. The method set forth in claim 34 wherein:

the encapsulator has a connection to the second network via a circuit in a public switched telephone network which is in turn coupled to a public modem pool belonging to the second packet network;

the connection is temporarily associated with an address in the second packet network; and

the source address is the address that is temporarily associated with the connection.

36. the method set forth in claim 35 further comprising the step of:

receiving the address that is temporarily associated with the connection from the second packet network and storing the address in the encapsulator as the source address.

37. the method set forth in any of claims 34 through 36 further comprising the step of:

receiving the destination address in the alternative from the first packet network or the second packet network and storing the destination address in the encapsulator.

38. A data storage device characterized in that:

the data storage device contains code which when executed in the encapsulator performs the method set forth in any of claims 34, 35, or 36.

39. A method used in a packet network of the type which transfers packets having headers which specify source addresses and destination addresses in the packet network to provide a packet to another packet network having different namespace,

the method comprising the steps of:

providing a temporary address in the packet network to a source of the packet which is external to the network;

receiving the packet from the source, the received packet having a header with the temporary address as a source address and the address of a decapsulator as a destination address;

routing the packet to the decapsulator; and

in the decapsulator, removing the header and providing the packet to the other packet network.

40. The method set forth in claim 39 further comprising the step of:

establishing a connection to the source of the packet which includes a modem in a public modem pool and a circuit in a public switched telephone system, the temporary address being assigned to the connection.

41. The method set forth in either of claims 39 or 40 further comprising the step of:

providing the address of the decapsulator to the source of the packet.

42. A method of encapsulating a packet received from a first packet network with a first namespace for transfer via a second packet network with a second namespace,
The method comprising the steps of:
receiving the packet from the first packet network; and
adding a header containing a source address and a destination address in the second packet network to the packet, the destination address being that of a decapsulator which the decapsulator operates to remove a header from a packet having the address of the decapsulator and provide the packet to the first packet network.

43. The method set forth in claim 42 further comprising the step of:
providing the packet with the header to the second second packet network via a connection which includes a modem coupled to a circuit in a public switched telephone network which is in turn coupled to a public modem pool belonging to the second packet network.

44. The method set forth in claim 43 wherein:
the second packet network temporarily associates an address in the second packet network with the circuit; and
the source address is the temporarily-associated address.

45. The method set forth in any of claims 42 through 44 further comprising the step of:
receiving the destination address from, in the alternative, the first packet network and the second packet network.

46. A data storage device characterized in that:
The data storage device contains code which, when executed on a processor, implements the method of any of claims 42, 43, or 44.

47. A method of responding interactively to a first packet received from a high-bandwidth forward channel belonging to a first packet network by providing a second packet to a return channel belonging to a second packet network accessible

via a circuit in a public switched telephone network, the first and second packet networks having different namespaces and the method comprising the steps performed by a processor of:

storing a first source address in the second packet network and a first destination address in the second packet network in memory accessible to the processor, the destination address being that of a decapsulator in the second packet network, the decapsulator operating to remove a header from a packet having the address of the decapsulator and provide the packet to the first packet network;

receiving the first packet via a first interface and providing the first packet to an interactive device via a second interface;

receiving the second packet via the second interface, the second packet being sent by the interactive device in response to the first packet and having a first header with a source and destination in the first packet network;

adding an additional header containing the first source address and the first destination address to the second packet; and

providing the second packet to the second packet network via a second interface coupled to the circuit in the public switched telephone network.

48. The method set forth in claim 47 further comprising the step of:

responding to a packet received via either the first or second packet network that contains a message specifying an address of a decapsulator by storing the address of the decapsulator in the memory as the first destination address.

49. The method set forth in claim 47 or claim 48 further comprising the step of:

receiving an address associated with the circuit from the second packet network and storing that address in memory as the first source address.

50. A data storage device characterized in that:

the data storage device contains code which when executed performs the method set forth in claim 47 or claim 48.

51. An interactive system comprising:

a first packet network which serves as a first channel, the first packet network having a first namespace;

a second packet network which serves as a second channel, the second packet network having a second namespace;

interactive means having an address in one of the packet networks for receiving a first packet of data from the one packet network and providing a second packet in response thereto which has a first destination in the one network;

an encapsulator which is coupled to the interactive means and has access to the other packet network for encapsulating the second packet with a header specifying a second destination in the other packet network and providing the packet to the other packet network; and

at the second destination, a decapsulator which has access to the one packet network for removing the header from the second packet and providing the second packet to the one packet network.

52. The interactive system set forth in claim 51 wherein:

the second packet network includes means for temporarily assigning a temporary address in the second packet network to a user thereof; and

the encapsulator means further specifies the temporary address as a source in the header.

53. The interactive system set forth in claim 52 further comprising:

a connection between the encapsulator means and the second packet network including a first modem, a circuit in a public switched telephone system, and a modem in a public modem pool which provides access to the second network; and

wherein the means for temporarily assigning assigns one of the temporary addresses to the connection and provides the assigned temporary address to the encapsulator via the connection.

54. The interactive system set forth in claim 51 wherein:

either the first packet network or the second packet network provides the second destination to the encapsulator.

55. The interactive system set forth in any of claims 51 through 54 wherein:

the first channel and the second channel have substantially differing bandwidths.

56. The interactive system set forth in claim 55 wherein:

one of the channels is a forward channel; and

another of the channels is a return channel.

57. The interactive system set forth in any of claims 51 through 54 wherein:

one of the channels is a forward channel; and

another of the channels is a return channel.

58. Improved apparatus for receiving first packets from a first channel belonging to one packet network and providing second packets to a second channel belonging to another packet network,

the apparatus including

a first interface to the first channel,

a second interface to the second channel,

a third interface to apparatus which receives the first packets and provides the second packets, and

control apparatus including a microprocessor and memory containing programs and data used by the microprocessor, the microprocessor operating under control of the programs and data to control the first, second, and third interfaces, and the improvement comprising:

a first source address in the other packet network and a first destination address in the other packet network stored in the memory, the destination address being that of a decapsulator in the other packet network, the decapsulator operating to remove a header from a packet having the address of the decapsulator and provide the packet to the one packet network; and

a first program stored in the memory means which, when executed by the microprocessor, causes an additional header containing the first source address and the first destination address to be added to a packet received from the third interface that is to be sent via the the second interface and the other packet network to a second destination address in the one network.

59. The apparatus set forth in claim 58 further comprising:

a second program stored in the memory which is executed by the microprocessor in response to a packet received via either the one or the other packet network that contains a message specifying an address of a decapsulator and which, when executed, stores the address of the decapsulator in the memory as the first destination address.

60. The apparatus set forth in claim 58 or claim 59 further comprising:

a third program which is executed by the microprocessor in response to establishment of the circuit in the switched public telephone network and which, when executed, receives an address associated with the circuit from the other packet network and stores that address in memory as the first source address.

61. The apparatus set forth in claim 58 or claim 59 wherein:

the first channel and the second channel have substantially different bandwidths.

62. The interactive system set forth in claim 61 wherein:

one of the channels is a forward channel; and

another of the channels is a return channel.

63. The interactive system set forth in claim 58 or 59 wherein:

one of the channels is a forward channel; and

another of the channels is a return channel.

64. A method of responding interactively to a first packet received from a first channel belonging to a first packet network by providing a second packet to a second channel belonging to a second packet network, the first and second packet networks having different namespaces and

the method comprising the steps performed by a processor of:

storing a first source address in one of the packet networks and a first destination address in the one second packet network in memory accessible to the processor, the destination address being that of a decapsulator in the one packet network, the decapsulator operating to remove a header from a packet having the address of the decapsulator and provide the packet to the other packet network;

receiving the first packet via a first interface and providing the first packet to an interactive device via a second interface;

receiving the second packet via the second interface, the second packet being sent by the interactive device in response to the first packet and having a first header with a source and destination in the other packet network;

adding an additional header containing the first source address and the first destination address to the second packet; and

providing the second packet to the one packet network.

65. The method set forth in claim 64 further comprising the step of:

responding to a packet received via either the first or second packet network that contains a message specifying an address of a decapsulator by storing the address of the decapsulator in the memory as the first destination address.

66. The method set forth in claim 64 or claim 65 further comprising the step of:

receiving an address associated with the circuit from the second packet network and storing that address in memory as the first source address.

67. The method set forth in claim 64 or claim 65 wherein:

the first channel and the second channel have substantially different bandwidths.

68. The method set forth in claim 67 wherein:

one of the channels is a forward channel; and

another of the channels is a return channel.

69. The method set forth in claim 64 or claim 65 wherein:

one of the channels is a forward channel; and

another of the channels is a return channel.

70. The method set forth in claim 64 or claim 65 wherein:

at least one of the packet networks is accessible via a circuit in a public switched telephone network.

71. A data storage device characterized in that:

the data storage device contains code which when executed performs the method set forth in claim 64 or claim 65.

FIG. 1
(PRIOR ART)

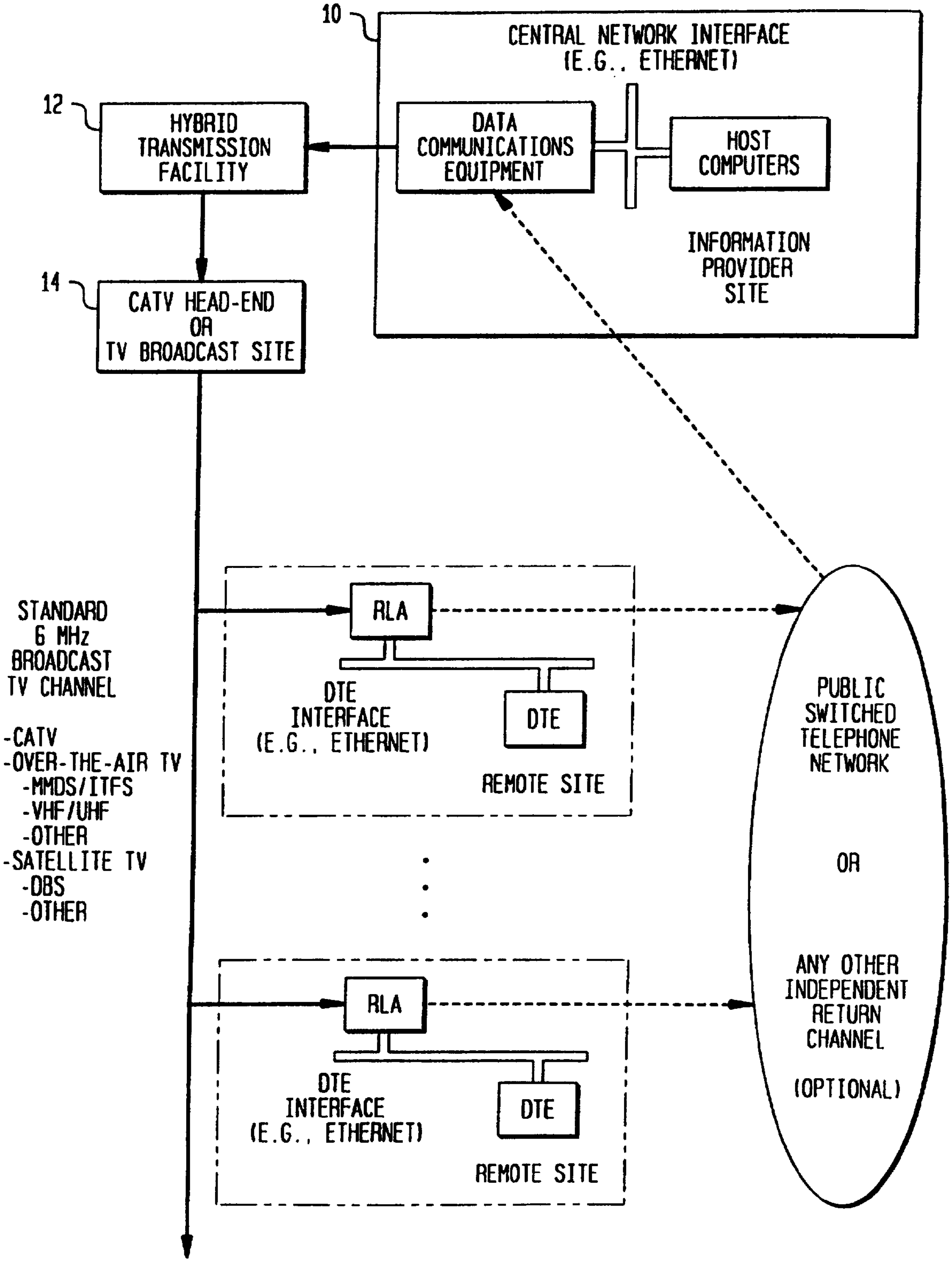

10
CENTRAL NETWORK INTERFACE
(E.G., ETHERNET)
12
HYBRID TRANSMISSION FACILITY
DATA COMMUNICATIONS EQUIPMENT
HOST COMPUTERS
INFORMATION PROVIDER SITE
14
CATV HEAD-END OR TV BROADCAST SITE
STANDARD 6 MHz BROADCAST TV CHANNEL
-CATV
-OVER-THE-AIR TV
-MMDS/ITFS
-VHF/UHF
-OTHER
-SATELLITE TV
-DBS
-OTHER
RLA
DTE INTERFACE (E.G., ETHERNET)
DTE
REMOTE SITE
RLA
DTE INTERFACE (E.G., ETHERNET)
DTE
REMOTE SITE
PUBLIC SWITCHED TELEPHONE NETWORK
OR
ANY OTHER INDEPENDENT RETURN CHANNEL
(OPTIONAL)

FIG. 2
(PRIOR ART)

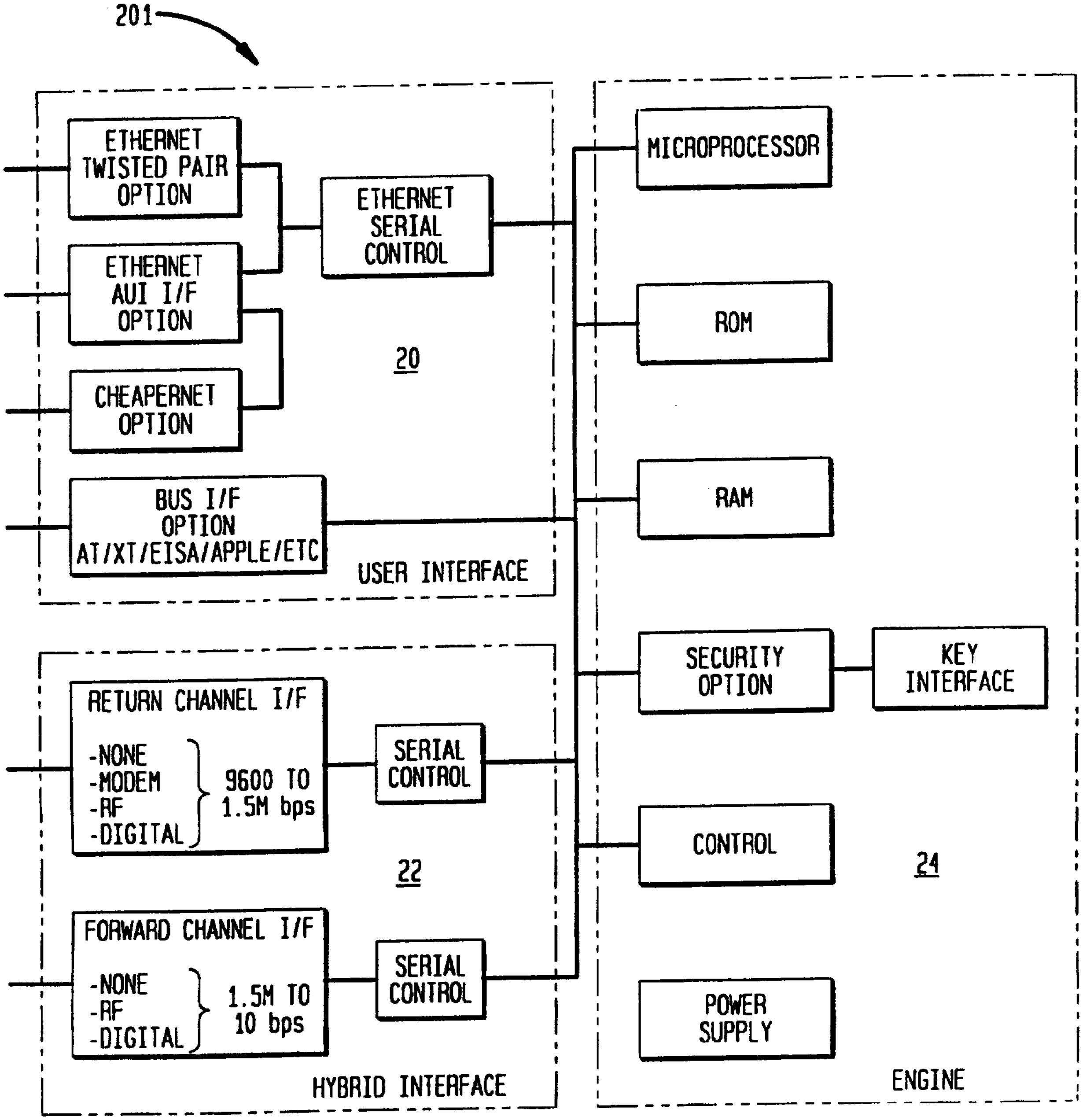


FIG. 3

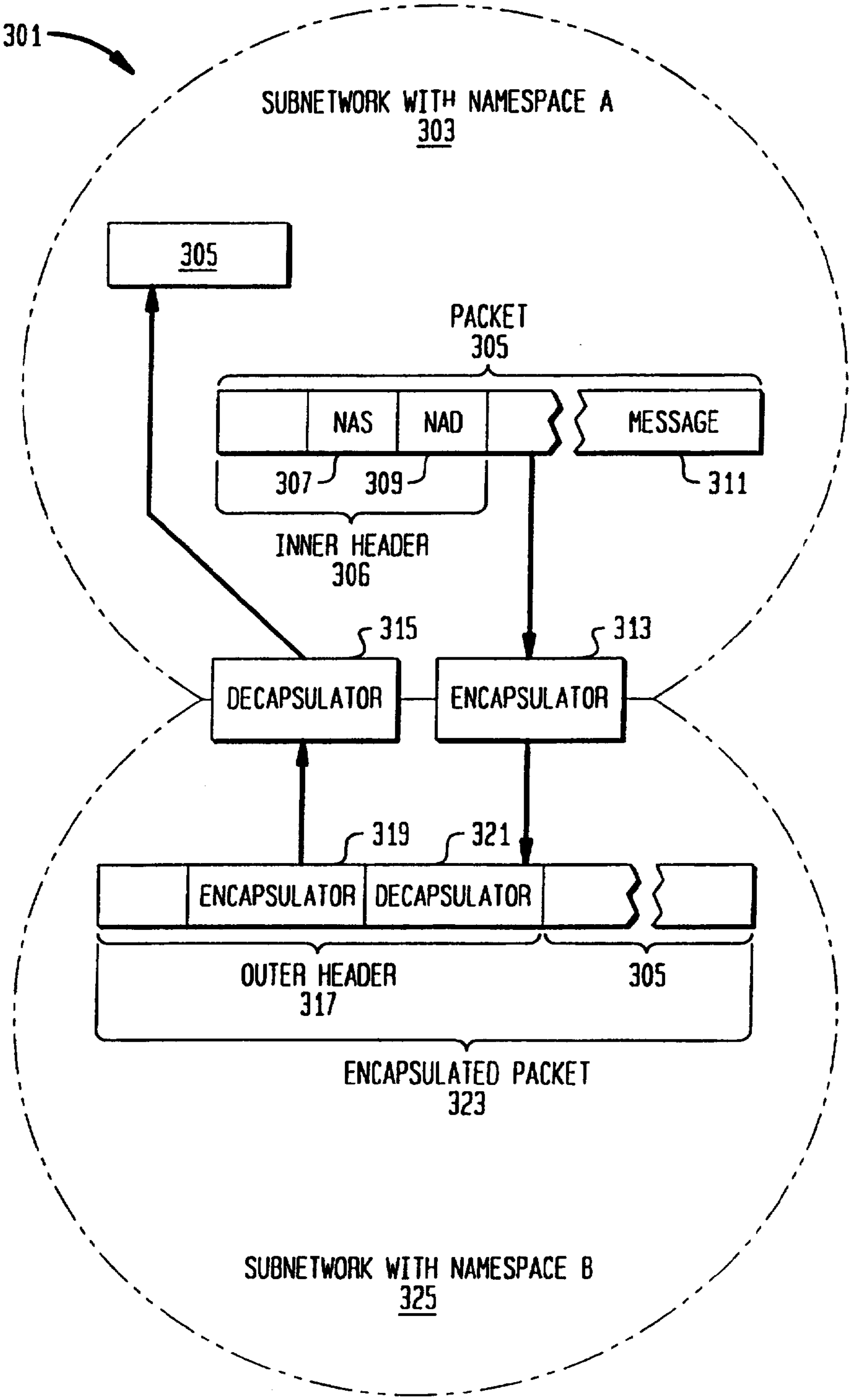

301
SUBNETWORK WITH NAMESPACE A
303
305
PACKET
305
NAS
NAD
MESSAGE
307
309
311
INNER HEADER
306
315
313
DECAPSULATOR
ENCAPSULATOR
319
321
ENCAPSULATOR
DECAPSULATOR
OUTER HEADER
317
305
ENCAPSULATED PACKET
323
SUBNETWORK WITH NAMESPACE B
325

FIG. 4

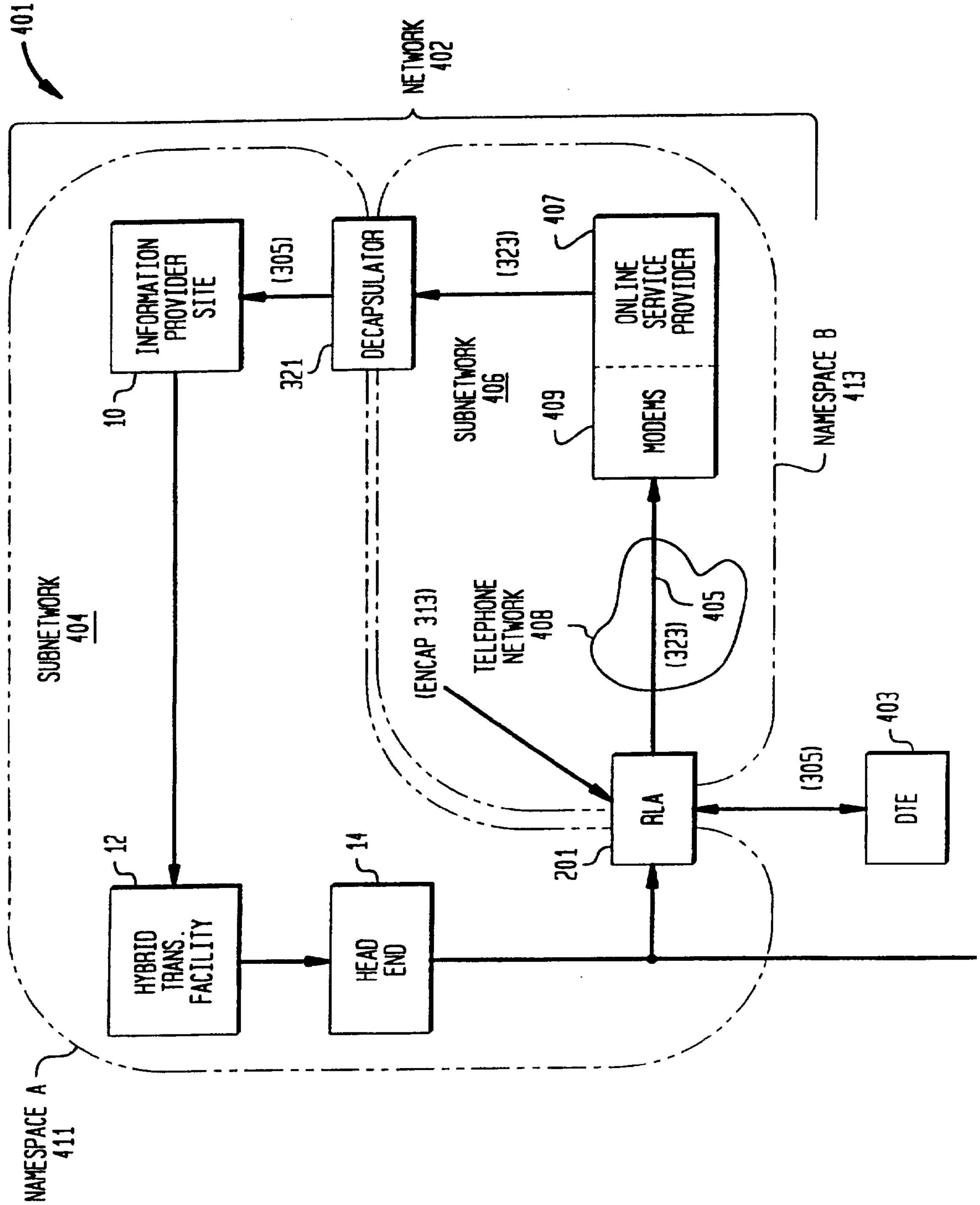
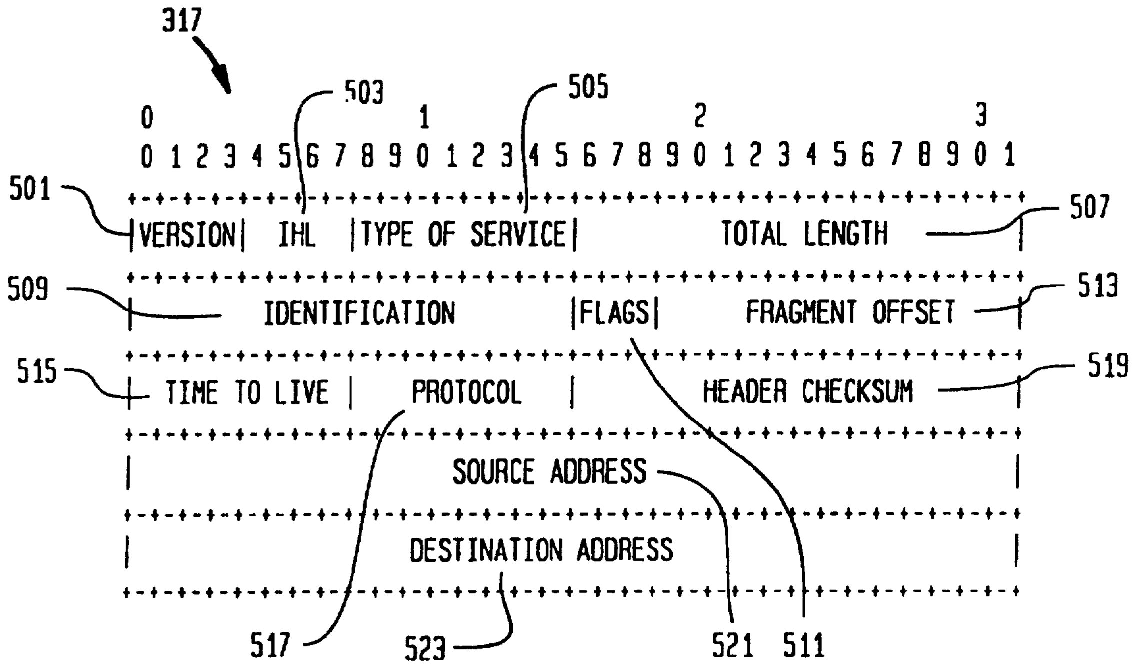

401
NETWORK 402
SUBNETWORK 404
INFORMATION PROVIDER SITE
10
(305)
DECAPSULATOR
321
(323)
407
ONLINE SERVICE PROVIDER
MODEMS
409
SUBNETWORK 406
NAMESPACE B 413
TELEPHONE NETWORK 408
(323)
405
(ENCAP 313)
RLA
201
(305)
DTE
403
HYBRID TRANS. FACILITY
12
HEAD END
14
NAMESPACE A 411

FIG. 5



THE FIELDS ARE AS FOLLOWS:

501 VERSION = 4
503 IHL (HEADER LENGTH) = 5
505 TYPE OF SERVICE: COPIED FROM USER'S PACKET
507 TOTAL LENGTH: SIZE OF USER'S PACKET + 20
509 IDENTIFICATION: NEW SEQUENCE NUMBER GENERATED
511 { FLAGS (BIT 0) = 0
FLAGS (BIT 1): 'Don't Fragment' COPIED FROM USER'S PACKET
FLAGS (BIT 2): 'More Fragments' SET AS NEEDED }
515 TIME TO LIVE = 255
517 PROTOCOL = 4 (IP IN IP ENCAPSULATION)
519 SOURCE: DIALUP ADDRESS
521 DESTINATON: DECAPSULATOR ADDRESS

FIG. 6

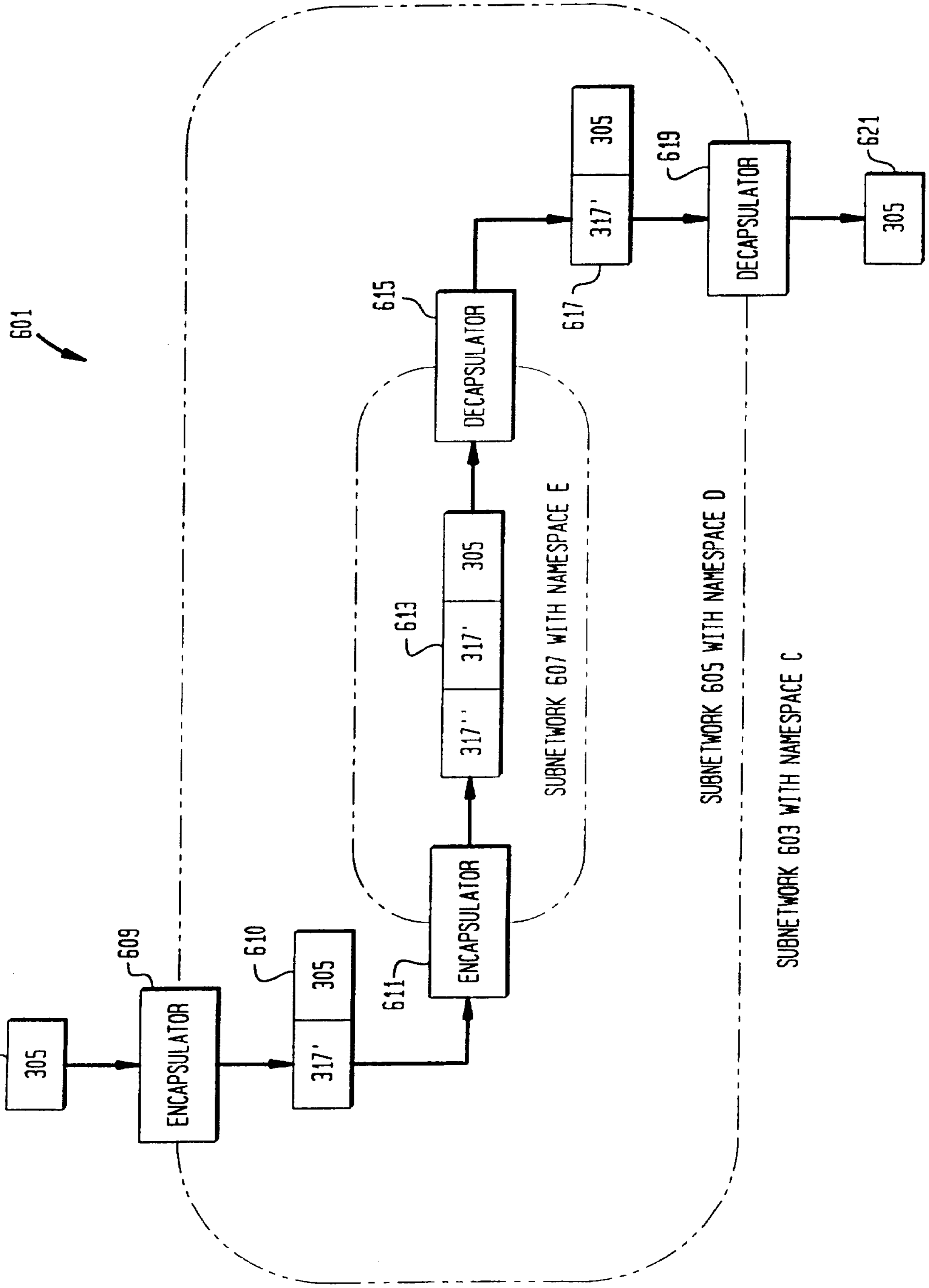

601
608
305
609
ENCAPSULATOR
610
317'
305
611
ENCAPSULATOR
613
317''
317'
305
DECAPSULATOR
615
617
317'
305
619
DECAPSULATOR
621
305
SUBNETWORK 607 WITH NAMESPACE E
SUBNETWORK 605 WITH NAMESPACE D
SUBNETWORK 603 WITH NAMESPACE C

FIG. 7

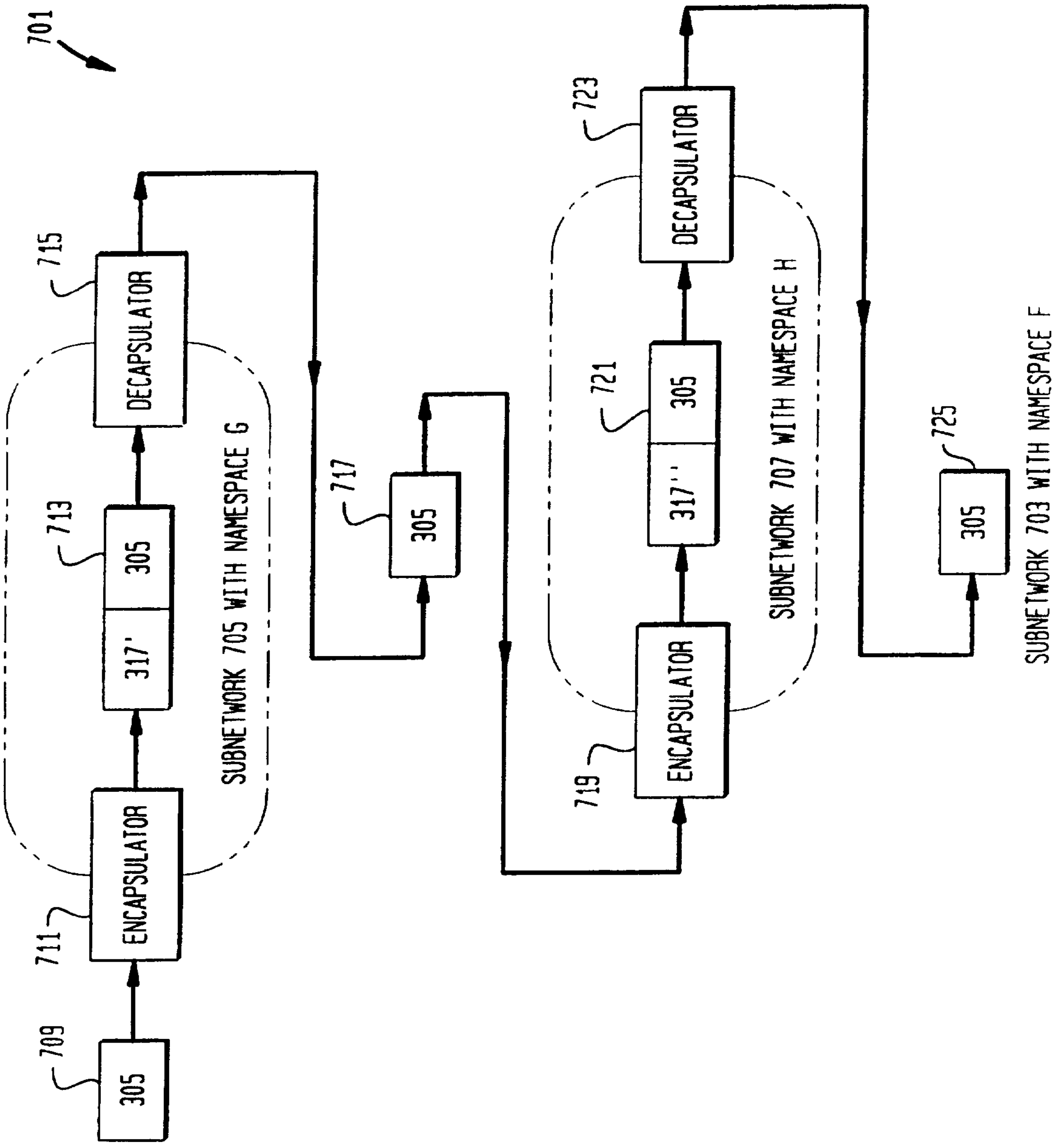

701
709
305
711
ENCAPSULATOR
713
317'
305
715
DECAPSULATOR
SUBNETWORK 705 WITH NAMESPACE G
717
305
719
ENCAPSULATOR
721
317''
305
723
DECAPSULATOR
SUBNETWORK 707 WITH NAMESPACE H
725
305
SUBNETWORK 703 WITH NAMESPACE F

FIG. 8

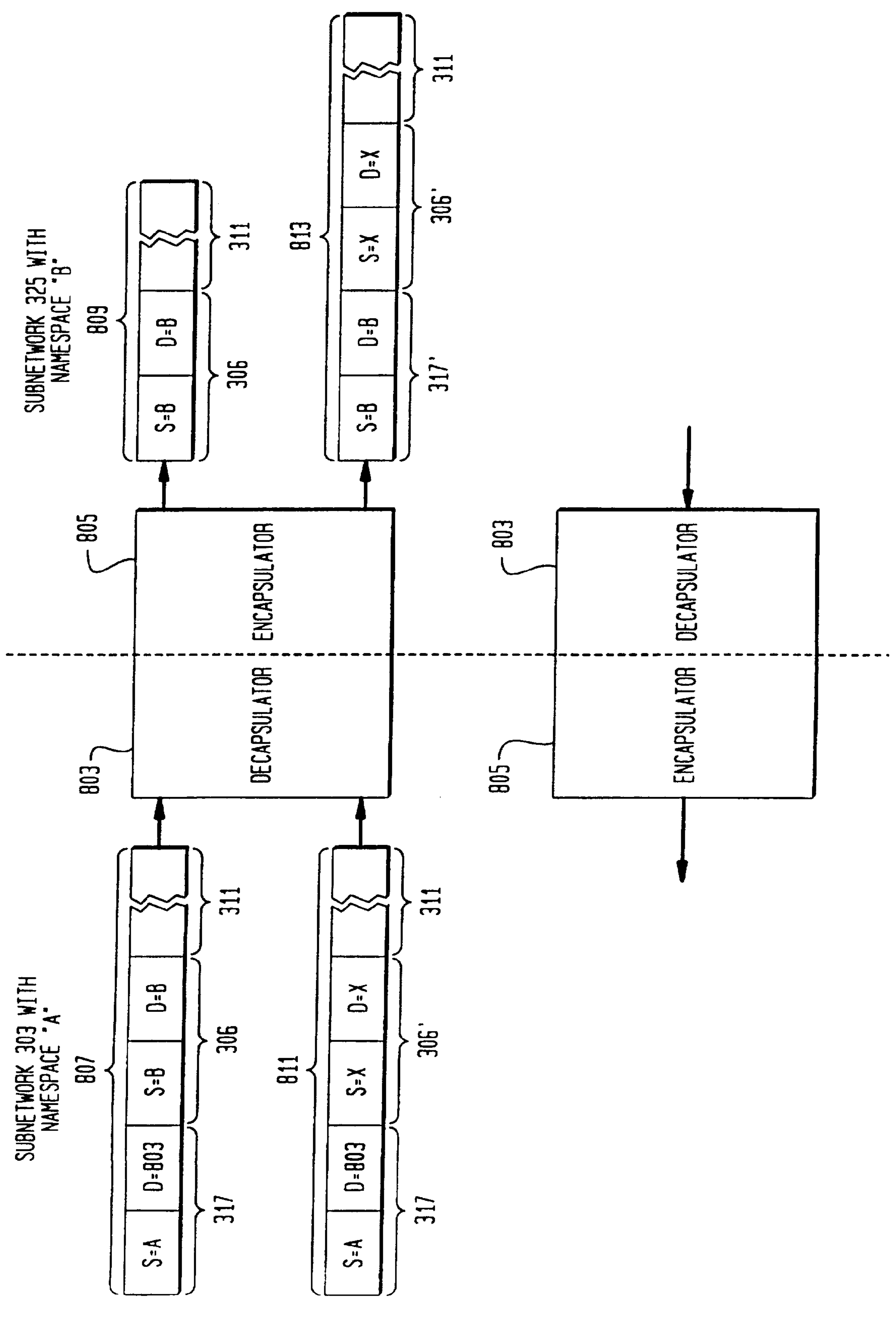

SUBNETWORK 303 WITH NAMESPACE "A"
807
S=A
D=803
S=B
D=B
317
306
311
811
S=A
D=803
S=X
D=X
317
306'
311
803
DECAPSULATOR
805
ENCAPSULATOR
SUBNETWORK 325 WITH NAMESPACE "B"
809
S=B
D=B
306
311
813
S=B
D=B
S=X
D=X
317'
306'
311
805
ENCAPSULATOR
803
DECAPSULATOR

FIG. 9

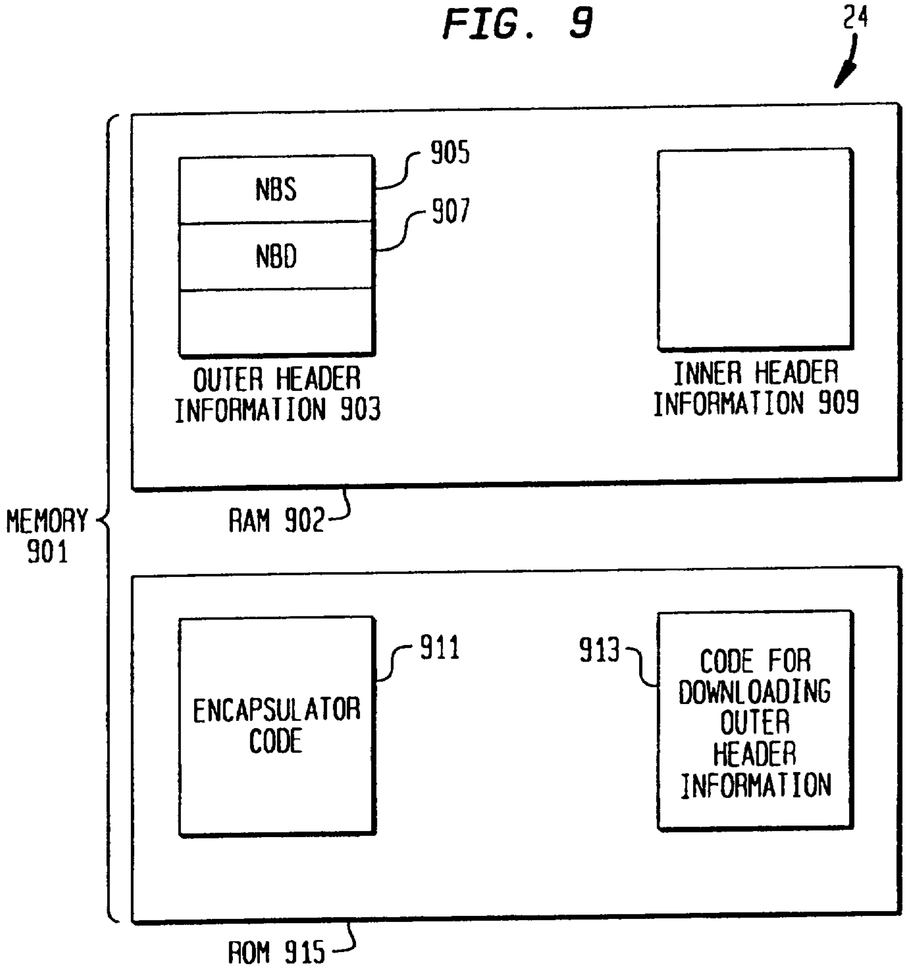
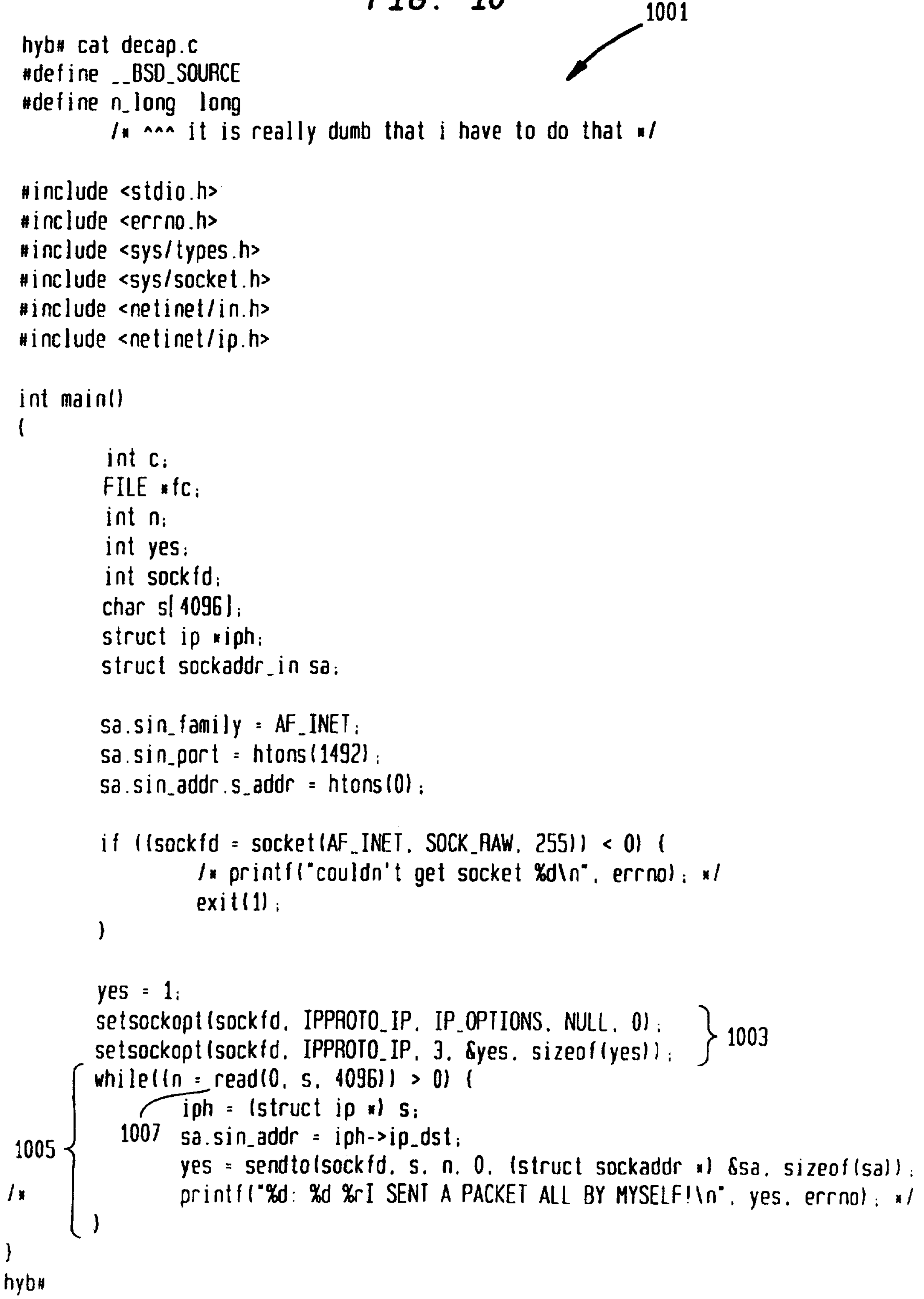

24
NBS
905
NBD
907
OUTER HEADER INFORMATION 903
INNER HEADER INFORMATION 909
RAM 902
MEMORY 901
ENCAPSULATOR CODE
911
913
CODE FOR DOWNLOADING OUTER HEADER INFORMATION
ROM 915

FIG. 10



```
hyb# cat decap.c
#define __BSD_SOURCE
#define n_long  long
        /* ^^^ it is really dumb that i have to do that */

#include <stdio.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/ip.h>

int main()
(
        int c;
        FILE *fc;
        int n;
        int yes;
        int sockfd;
        char s[4096];
        struct ip *iph;
        struct sockaddr_in sa;

        sa.sin_family = AF_INET;
        sa.sin_port = htons(1492);
        sa.sin_addr.s_addr = htons(0);

        if ((sockfd = socket(AF_INET, SOCK_RAW, 255)) < 0) (
                /* printf("couldn't get socket %d\n", errno); */
                exit(1);
        )

        yes = 1;
        setsockopt(sockfd, IPPROTO_IP, IP_OPTIONS, NULL, 0);
        setsockopt(sockfd, IPPROTO_IP, 3, &yes, sizeof(yes));
        while((n = read(0, s, 4096)) > 0) (
                iph = (struct ip *) s;
                sa.sin_addr = iph->ip_dst;
                yes = sendto(sockfd, s, n, 0, (struct sockaddr *) &sa, sizeof(sa));
/*              printf("%d: %d %rI SENT A PACKET ALL BY MYSELF!\n", yes, errno); */
        )
)
hyb#
```

INTERNATIONAL SEARCH REPORT

Interi nal Application No
PCT/US 97/13943

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04L12/66

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	W. SIMPSON: "IP in IP Tunneling" NETWORK WORKING GROUP, no. 1853, October 1995, HTTP://WWW.IT.KTH.SE/DOCS/RFCS/RFC1853.TXT , XP002051505 cited in the application see the whole document ---	1-71
X	LIBA SVOBODOVA ET AL: "HETEROGENEITY AND OSI" IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, vol. 8, no. 1, 1 January 1990, pages 67-79, XP000133533 see abstract see page 75, paragraph B --- -/--	1-71

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publicationdate of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

3

Date of the actual completion of theinternational search 14 January 1998	Date of mailing of the international search report 28/01/1998
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Adkhis, F

INTERNATIONAL SEARCH REPORT

Inter ɔnal Application No
PCT/US 97/13943

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	WO 96 21983 A (NOKIA TELECOMMUNICATIONS OY ;AHOPELTO JUHA PEKKA (FI); KARI HANNU) 18 July 1996 see abstract see page 2, line 22 - line 25 see page 3, line 9 - line 30 see page 4, line 20 - page 5, line 10 ---	1-71
A	EP 0 700 231 A (AT & T CORP) 6 March 1996 see abstract see page 3, column 3, line 14 - line 46 ---	1-71
A	NORITOSHI DEMIZU ET AL: "DDT - A VERSATILE TUNNELING TECHNOLOGY" COMPUTER NETWORKS AND ISDN SYSTEMS, vol. 27, no. 3, 1 December 1994, pages 493-502, XP000483281 see page 493, paragraph 1 - page 495, line 2.2 see page 496, paragraph 3.2 ---	1-71
A	BELLOVIN S M ET AL: "NETWORK FIREWALLS" IEEE COMMUNICATIONS MAGAZINE, vol. 32, no. 9, 1 September 1994, pages 50-57, XP000476555 see page 56, left-hand column, line 60 - right-hand column, line 70 -----	1-71

3

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inte onal Application No

PCT/US 97/13943

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9621983 A	18-07-96	FI 950116 A	11-07-96
		AU 4392896 A	31-07-96
		CA 2209715 A	18-07-96
		EP 0804844 A	05-11-97
		NO 973177 A	09-09-97
EP 0700231 A	06-03-96	US 5623605 A	22-04-97
		CN 1122979 A	22-05-96
		JP 8111693 A	30-04-96